REBEL: Reinforcement Learning via Regressing Relative Rewards

Anonymous authors
Paper under double-blind review

Abstract

While originally developed for continuous control problems, Proximal Policy Optimization (PPO) has emerged as the work-horse of a variety of reinforcement learning (RL) applications including the fine-tuning of generative models. Unfortunately, PPO requires multiple heuristics to enable stable convergence (e.g. value networks, clipping) and is notorious for its sensitivity to the precise implementation of these components. In response, we take a step back and ask what a *minimalist* RL algorithm for the era of generative models would look like. We propose `REBEL`, an algorithm that cleanly reduces the problem of policy optimization to regressing the *relative reward* between two completions to a prompt in terms of the policy, enabling strikingly lightweight implementation. In theory, we prove that fundamental RL algorithms like Natural Policy Gradient can be seen as variants of `REBEL`, which allows us to match the strongest known theoretical guarantees in terms of convergence and sample complexity in the RL literature. `REBEL` can also cleanly incorporate offline data and handle the intransitive preferences we frequently see in practice. Empirically, we find that `REBEL` provides a unified approach to language modeling and image generation with stronger or similar performance as PPO and DPO, all while being simpler to implement and more computationally tractable than PPO.

1 Introduction

The generality of the reinforcement learning (RL) paradigm is striking: from continuous control problems (Kalashnikov et al., 2018) to, recently, the fine-tuning of generative models (Stiennon et al., 2022; Ouyang et al., 2022), RL has enabled concrete progress across a variety of decision-making tasks. Specifically, when it comes to fine-tuning generative models, Proximal Policy Optimization (PPO, Schulman et al. (2017)) has emerged as the de-facto RL algorithm of choice, from language models (LLMs) (Ziegler et al., 2020; Stiennon et al., 2022; Ouyang et al., 2022; Touvron et al., 2023) to image generative models (Black et al., 2023; Fan et al., 2024; Oertell et al., 2024).

If we take a step back however, it is odd that we are using an algorithm designed for optimizing two-layer networks for continuous control tasks from scratch for fine-tuning the billions of parameters of modern-day generative models. In the continuous control setting, the randomly initialized neural networks and the possible stochasticity in the dynamics necessitate variance reduction through a learned value function as a baseline (Schulman et al., 2015b), while clipping updates is important to limit distribution shift from iteration to iteration (Kakade & Langford, 2002). This means that when applied to generative model fine-tuning, we need to store four models in memory simultaneously (the policy, the reference policy, the critic, and the reward model), each with billions of parameters. Furthermore, we often add a KL regularization to the base model for fine-tuning, making explicit clipping unnecessary nor advisable, as pointed out by Ahmadian et al. (2024). Even outside of the generative modeling context, PPO is notorious for the wide range of performances measured, with differences being attributed to seemingly inconsequential implementation details (Henderson et al., 2019; Engstrom et al., 2020). This begs the question: ***Are there simpler algorithms that scale to modern RL applications?***

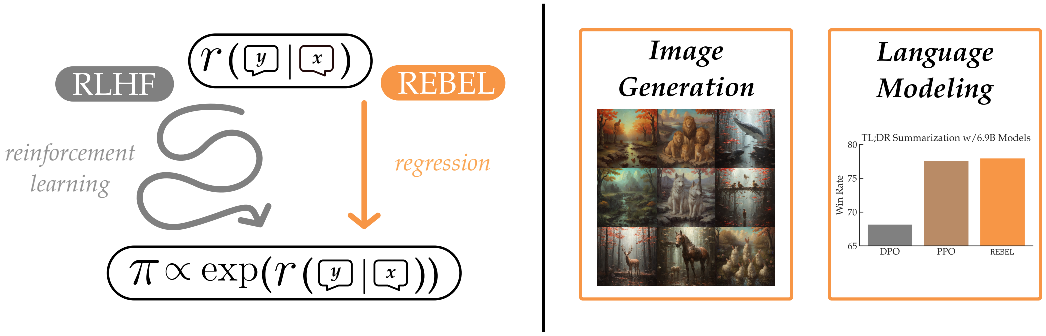


Figure 1: We present REBEL: a simple and scalable RL algorithm that performs policy optimization via *iteratively regressing the difference in rewards in terms of the policy*, allowing us to eliminate much of the complexity (e.g. value functions, clipping) of algorithms like PPO (Schulman et al., 2017). We apply REBEL to problems in both image generation and language modeling and find that despite its conceptual and implementation-level simplicity, REBEL is able to match or sometimes outperform the performance of PPO while out-performing purely offline techniques like DPO (Rafailov et al., 2023).

Our answer is REBEL: an algorithm that *reduces the problem of RL to solving a sequence of squared loss regression problems* on iteratively collected datasets. The regression problems directly use policies to predict the difference in rewards. This allows us to eliminate the complexity of value functions, avoid heuristics like clipping, and scale easily to problems in both language modeling and image generation. Our key insight is that ***a regressor that can predict the difference in rewards between trajectories in a dataset implicitly captures an improved policy.***

Rather than being a heuristic, REBEL comes with strong guarantees in theory and can be seen as a strict generalization of classical techniques (e.g., NPG) in reinforcement learning. Furthermore, REBEL cleanly incorporates offline datasets when available, can be extended to robustly handle intransitive preferences (Swamy et al., 2024), and empirically out-performs techniques like PPO and DPO (Rafailov et al., 2023) in language generation and has a faster convergence with a similar asymptotic performance in image generation.

2 REBEL: REgression to RElative REward Based RL

2.1 Notation

We consider the Contextual Bandit formulation (Langford & Zhang, 2007) of RL which has been used to formalize the generation process of models like LLMs (Rafailov et al., 2023; Ramamurthy et al., 2022; Chang et al., 2023) and Diffusion Models (Black et al., 2023; Fan et al., 2024; Oertell et al., 2024) due to the determinism of the transitions. More explicitly, in the deterministic transition setting, explicit states are not required as they can be equivalently represented by a sequence of actions. Furthermore, the entire sequence of actions can be considered as a single "arm" in a bandit problem with an exponentially large action space.

We denote by (x, y) a prompt/response pair with $x \in \mathcal{X}$ as a prompt and $y \in \mathcal{Y}$ as a response (e.g., a sequence of tokens, or in general a sequence of actions). We assume access to a reward function $r(x, y)$ from which we can query for reward signals (the exact form of r does not need to be known). Querying r at (x, y) will return a scalar $r(x, y)$ measuring the quality of the response. Such a reward function could be a pre-defined metric (e.g., Rouge score against human responses) or it could be learned from an offline human demonstration or preference data (e.g., the RLHF paradigm (Christiano et al., 2017; Ziegler et al., 2020)), as explored in our experiments.

Denote by $\pi \in \mathcal{X} \mapsto \Delta(Y)$, a policy (e.g. LLM) that maps from a prompt x to a distribution over the response space $\mathcal{Y}$. We use ρ to denote the distribution over prompts (i.e. initial states / contexts) x. Throughout the paper, we use $\pi_\theta(y|x)$ to denote a parameterized policy with parameter θ (e.g., a neural network policy). At times we interchangeably use π_t and π_{θ_t} when it is clear from the context.

Algorithm 1 REgression to RElative REward Based RL (`REBEL`)

1: **Input**: Reward r, policy class $\Pi = \{\pi_\theta\}$, base distribution μ, learning rate η
2: Initialize policy π_{θ_0}.
3: **for** $t = 0$ to $T-1$ **do**
4: // Base distribution μ can either be an offline dataset or π_t.
5: Collect dataset $\mathcal{D}_t = \{x, y, y'\}$ where $x \sim \rho, y \sim \pi_t(\cdot|x), y' \sim \mu(\cdot|x)$
6: Solve square loss regression problem:

$$\theta_{t+1} = \underset{\theta}{\text{argmin}} \sum_{(x,y,y') \in \mathcal{D}_t} \left(\frac{1}{\eta} \left(\ln \frac{\pi_\theta(y|x)}{\pi_{\theta_t}(y|x)} - \ln \frac{\pi_\theta(y'|x)}{\pi_{\theta_t}(y'|x)} \right) - (r(x,y) - r(x,y')) \right)^2 \quad (1)$$

7: **end for**

We emphasize that while we focus on the bandit formulation for notation simplicity, the algorithms proposed here can be applied to *any* deterministic MDP where x is the initial state and the trajectory y consists of the sequence of actions.

At each iteration of all algorithms, our goal will be to solve the following KL-constrained RL problem:

$$\pi_{t+1} = \underset{\pi}{\text{argmax}}\, \mathbb{E}_{x,y\sim\pi(\cdot|x)} r(x,y) - \frac{1}{\eta} \mathbb{E}_x \text{KL}\left(\pi(\cdot|x) || \pi_t(\cdot|x)\right). \quad (2)$$

Intuitively, this can be thought of asking for the optimizer to fine-tune the policy π_{t+1} according to r while staying close to some baseline policy π_t.

2.2 Deriving `REBEL`: REgression to RElative REward Based RL

From Ziebart et al. (2008), we know that there exists a closed-form solution to the above *minimum relative entropy* problem (Eq. 2, Grünwald & Dawid (2004)):

$$\forall x, y : \pi_{t+1}(y|x) = \frac{\pi_t(y|x)\exp(\eta r(x,y))}{Z(x)}; \; Z(x) = \sum_y \pi_t(y|x)\exp(\eta r(x,y)). \quad (3)$$

As first pointed out by Rafailov et al. (2023), observe that we can invert Eq. 3 and write the reward as a function of the policy, i.e. the "DPO Trick":

$$\forall x, y : r(x,y) = \frac{1}{\eta}\left(\ln(Z(x)) + \ln\left(\frac{\pi_{t+1}(y|x)}{\pi_t(y|x)}\right)\right). \quad (4)$$

As soon as $\mathcal{X}$ and $\mathcal{Y}$ become large, we can no longer guarantee the above expression holds exactly at all (x,y) and therefore need to turn our attention to choosing a policy such that Eq. 4 is approximately true. We propose using a simple *square loss* objective between the two sides of Eq. 4 to measure the goodness of a policy, i.e. reducing RL to a regression problem: $\left(r(x,y) - \frac{1}{\eta}\left(\ln(Z(x)) + \ln\left(\frac{\pi_{t+1}(y|x)}{\pi_t(y|x)}\right)\right)\right)^2$. Unfortunately, this loss function includes the *partition function* $Z(x)$, which can be challenging to approximate over large input / output domains. However, observe that $Z(x)$ only depends on x and not y. Thus, if we have access to *paired samples*, i.e. (x,y) and (x,y'), we can instead regress the *difference in rewards* to eliminate this term from our objective:

$$\left((r(x,y) - r(x,y')) - \frac{1}{\eta}\left(\ln\left(\frac{\pi_{t+1}(y|x)}{\pi_t(y|x)}\right) - \ln\left(\frac{\pi_{t+1}(y'|x)}{\pi_t(y'|x)}\right)\right)\right)^2. \quad (5)$$

Of course, we need to evaluate this loss function on some distribution of samples. In particular, we propose using an on-policy dataset $\mathcal{D}_t = \{x, y, y'\}$ with $x \sim \rho, y \sim \pi_t(\cdot|x), y' \sim \mu(\cdot|x)$, where μ is some *base distribution*. The base distribution μ can either be a fixed offline dataset (e.g. the instruction fine-tuning dataset) or π_t itself. Thus, the choice of base distribution μ determines whether REBEL is hybrid or fully online. Putting it all together, we arrive at our core REBEL objective in Eq. 1. Critically, observe that if we were able to perfectly solve this regression problem, we would indeed recover the optimal solution to the KL-constrained RL problem we outlined in Eq. 2.

3 Understanding REBEL as an Adaptive Policy Gradient

In this section, we interpret REBEL as an adaptive policy gradient methods and illuminate its strengths over past techniques. We include a summary of the related work in Appendix A.

3.1 Adaptive Gradient Algorithms for Policy Optimization

Mirror Descent. If $\mathcal{X}$ and $\mathcal{Y}$ are small discrete spaces, we can used the closed-form expression for the minimum relative entropy problem (Eq. 3). This is equivalent to the classic Mirror Descent (MD, Ziebart et al. (2008)) algorithm with KL as the Bregman divergence.

Natural Policy Gradient. When $\mathcal{Y}$ and $\mathcal{X}$ are large, we use a parameterized policy denoted as π_θ with parameter θ. Natural Policy Gradient (NPG, Kakade (2001)) approximates the KL in Equation 2 via its second-order Taylor expansion, whose Hessian is known as the Fisher Information Matrix (FIM, Bagnell & Schneider (2003)), F_t, i.e. $F_t = \mathbb{E}_{x,y\sim\pi_{\theta_t}(\cdot|x)}\left[\nabla \ln \pi_{\theta_t}(y|x) \nabla \ln \pi_{\theta_t}(y|x)^\top\right]$ and $\mathbb{E}_x \mathrm{KL}(\pi_\theta(\cdot|x)||\pi_{\theta_t}(\cdot|x)) \approx (\theta-\theta_t)^\top F_t(\theta-\theta_t)$. The NPG update can be formulated as:

$$\theta_{t+1} = \theta_t + \eta F_t^\dagger \left(\mathbb{E}_{x,y\sim\pi_{\theta_t}(\cdot|x)} \nabla \ln \pi_{\theta_t}(y|x) r(x,y)\right) \tag{6}$$

where $F_t^\dagger$ is pseudo-inverse of F_t.

Proximal Policy Optimization. NPG, unfortunately, does not scale to modern generative models due to the need for computing the Fisher matrix inverse. Proximal Policy Optimization (PPO, Schulman et al. (2017)) takes a more direct route and uses clipped updates

$$\theta_{t+1} := \operatorname*{argmax}_\theta \mathbb{E}_{x,y\sim\pi_{\theta_t}(\cdot|x)} \mathrm{clip}\left(\frac{\pi_\theta(y|x)}{\pi_{\theta_t}(y|x)}; 1-\epsilon, 1+\epsilon\right) r(x,y). \tag{7}$$

While the clipping operator can set the gradient to be zero at samples (x,y) where $\pi_{\theta_{t+1}}(y|x)$ is much larger or smaller than $\pi_{\theta_t}(y|x)$, it cannot actually guarantee $\pi_{\theta_{t+1}}$ staying close to π_{θ_t}, a phenomenon empirically observed in prior work (Hsu et al., 2020). Furthermore, hard clipping is not adaptive – it treats all (x,y) equally and clips whenever the ratio is outside of a fixed range. In contrast, constraining the KL divergence to the prior policy allows one to vary the ratio $\pi(y|x)/\pi_t(y|x)$ at different (x,y), as long as the total KL divergence across the state space is small. Lastly, clipping reduces the effective size of a batch of training examples and thus wastes training samples.

3.2 Connections between REBEL and MD / NPG

Exact REBEL is Mirror Descent. First, to build intuition, we interpret our algorithm's behavior under the assumption that the least square regression optimization returns the exact Bayes Optimal solution (i.e., our learned predictor achieves zero prediction error everywhere):

$$\forall x, y, y': \quad \frac{1}{\eta}\left(\ln \frac{\pi_{\theta_{t+1}}(y|x)}{\pi_{\theta_t}(y|x)} - \ln \frac{\pi_{\theta_{t+1}}(y'|x)}{\pi_{\theta_t}(y'|x)}\right) = r(x,y) - r(x,y') \tag{8}$$

Conditioned on Eq. 8 being true, a few lines of algebraic manipulation reveals that there must exist a function $c(x)$ which is independent of y, such that $\forall x, y: \frac{1}{\eta} \ln \frac{\pi_{\theta_{t+1}}(y|x)}{\pi_{\theta_t}(y|x)} = r(x,y) + c(x)$. Taking an exp on both sides and re-arrange terms, we get $\forall x, y: \pi_{\theta_{t+1}}(y|x) \propto \pi_{\theta_t}(y|x) \exp(\eta r(x,y))$. In other words, under the strong assumption that least square regression returns a point-wise accurate estimator (i.e., Eq. 8), we see the REBEL recovers the exact MD update, which gives it *(a)* a fast $1/T$ convergence rate (Shani et al., 2020; Agarwal et al., 2021a), *(b)* conservativity, i.e., $\max_x \mathrm{KL}(\pi_{t+1}(\cdot|x)||\pi_t(\cdot|x))$ is bounded as long as $\max_{x,y}|r(x,y)|$ is bounded, and *(c)* monotonic policy improvement via the NPG standard analysis (Agarwal et al., 2021a).

NPG is Approximate REBEL with Gauss-Newton Updates. We provide another interpretation of REBEL by showing that NPG (Eq. 6) can be understood as a special case of REBEL where the least square problem in Eq. 1 is approximately solved via a single iteration of the Gauss-Newton algorithm.

As for any application of Gauss-Newton, we start by approximating our predictor $\frac{1}{\eta}\ln \pi_\theta(y|x)/\pi_{\theta_t}(y|x)$ by its first order Taylor expansion at θ_t: $\frac{1}{\eta}\left(\ln \pi_\theta(y|x) - \ln \pi_{\theta_t}(y|x)\right) \approx \frac{1}{\eta}\nabla_\theta \ln \pi_{\theta_t}(y|x)^\top(\theta-\theta_t)$, where $\approx$ indicates that we ignore higher order terms in the expansion. If we $\delta := \theta - \theta_t$ and replace $\frac{1}{\eta}\left(\ln \pi_\theta(y|x) - \ln \pi_{\theta_t}(y|x)\right)$ by its first order approximation in Eq. 1, we arrive at:

$$\min_\delta \mathbb{E}_{x\sim\rho, y\sim\pi_{\theta_t}(\cdot|x), y'\sim\mu(\cdot|x)}\left(\frac{1}{\eta}\left(\nabla_\theta \ln \pi_{\theta_t}(y|x) - \nabla_\theta \ln \pi_{\theta_t}(y'|x)\right)^\top \delta - \left(r(x,y) - r(x,y')\right)\right)^2. \quad (9)$$

Further simplifying notation, we denote the uniform mixture of π_t and μ as $\pi_{mix}(\cdot|x) := (\pi_t(\cdot|x) + \mu(\cdot|x))/2$ and the Fisher information matrix F_t averaged under said mixture as $F_t = \mathbb{E}_{x\sim\rho, y\sim\pi_{mix}(\cdot|x)}\left[\nabla_\theta \ln \pi_{\theta_t}(y|x)\left(\nabla_\theta \ln \pi_{\theta_t}(y|x)\right)^\top\right]$. Solving the above least square regression to obtain a minimum norm solution, we have the following claim.

Claim 1. *The minimum norm minimizer $\delta^\star$ of the least squares problem in Eq. 9 recovers an advantage-based NPG update: $\delta^\star := \eta F_t^\dagger\left(\mathbb{E}_{x\sim\rho, y\sim\pi_{mix}(\cdot|x)}\nabla_\theta \ln \pi_{\theta_t}(y|x)[A^{\pi_t}(x,y)]\right)$ where $F_t^\dagger$ is pseudo-inverse of F_t, and the* advantage *is defined as $A^{\pi_t}(x,y) := r(x,y) - \mathbb{E}_{y'\sim\pi_t(\cdot|x)} r(x,y)$.*

The proof of this claim is deferred to Appendix B. Observe that in REBEL, we never explicitly compute the advantage A^{π_t}. However, applying Gauss-Newton to our objective leads to an advantage-based NPG (rather than the traditional Q-function based NPG, e.g., Q-NPG from Agarwal et al. (2021a; 2019)) which indicates that predicting reward difference has an *implicit variance reduction* effect, as by definition, an advantage function includes a value function baseline. [1]

A REBEL *With* a Cause. Our algorithm REBEL addresses the limitations of NPG (scalability) and PPO (lack of conservativity or adaptivity) from above. First, unlike NPG, it does not rely on the Fisher information matrix at all and can easily scale to modern LLM applications, yet (as we will discuss below) can be interpreted as a *generalization* of NPG. Second, in contrast to PPO, it doesn't have unjustified heuristics and thus enjoys strong convergence and regret guarantees just like NPG. We also show how to extend REBEL to general preferences in Appendix C.

4 Theoretical Analysis

In the previous section, we interpret REBEL as the exact MD and show its convergence by assuming that least square regression always returns a predictor that is accurate *everywhere*. In this section, we significantly relax this condition via a reduction-based analysis. Formally, we assume the following generalization condition holds on the regressors we find.

Assumption 1 (Regression generalization bounds)**.** *Over T iterations, assume that for all t, we have the following for some ϵ:*

$$\mathbb{E}_{x\sim\rho, y\sim\pi_t(\cdot|x), y'\sim\mu(\cdot|x)}\left(\frac{1}{\eta}\left(\ln\frac{\pi_{\theta_{t+1}}(y|x)}{\pi_{\theta_t}(y|x)} - \ln\frac{\pi_{\theta_{t+1}}(y'|x)}{\pi_{\theta_t}(y'|x)}\right) - \left(r(x,y) - r(x,y')\right)\right)^2 \le \epsilon,$$

More justifications of the assumption is in Appendix D.

[1] Note that the original form of NPG is on-policy (Kakade, 2001), i.e., the expectations under π_t. Our formulation is more general: when set $\mu = \pi_t$, a Gauss-Newton step will recover the original on-policy form of NPG from Kakade (2001). More recent works have extended NPG beyond on-policy (e.g., Agarwal et al. (2021a; 2020)).

Data Coverage. Recall that the base distribution μ can be some behavior policy, which in RLHF can be a human labeler, a supervised fine-tuned policy (SFT), or just the current learned policy (i.e., on-policy). Given a test policy π, we denote by $C_{\mu\to\pi}$ the concentrability coefficient, i.e.

$$C_{\mu\to\pi} = \max_{x,y} \frac{\pi(y|x)}{\mu(y|x)}. \tag{10}$$

We say μ *covers* π if $C_{\mu\to\pi} < +\infty$. Our goal is to bound the regret between our learned policies and an arbitrary comparator π^* (e.g. the optimal policy if it is covered by μ) using ϵ and the concentrability coefficient defined in Eq. 10. The following theorem formally states the regret bound of our algorithm.

Theorem 1. *Under Assumption 1, after T many iterations, with a proper learning rate η, among the learned policies $\pi_1, \ldots, \pi_T$, there must exist a policy $\hat{\pi}$, such that:*

$$\forall \pi^* : \; \mathbb{E}_{x\sim\rho, y\sim\pi^*(\cdot|x)} r(x,y) - \mathbb{E}_{x\sim\rho, y\sim\hat{\pi}(\cdot|x)} r(x,y) \leq O\left(\sqrt{\frac{1}{T}} + \sqrt{C_{\mu\to\pi^*}\epsilon.}\right).$$

The above theorem shows a ***reduction from RL to supervised learning*** — as long as supervised learning works (i.e., ϵ is small), then REBEL can compete against any policy π^* that is covered by the base data distribution μ. In the regret bound, the $1/\sqrt{T}$ comes from Mirror Descent style update, and $C_{\mu\to\pi^*}\epsilon$ captures the cost of distribution shift: we train our regressors under distribution π_t and μ, but we want the learned regressor to predict well under π^*. Similar to the NPG analysis from Agarwal et al. (2021a), we now have a slower convergence rate $1/\sqrt{T}$, due to the fact that we have approximation error from learning. Such an agnostic regret bound — being able compete against any policy that is covered by training distributions – is the **strongest type of agnostic learning results known in the RL literature**, matching the best of what has appeared in prior policy optimization work including PSDP (Bagnell et al., 2003), CPI (Kakade & Langford, 2002), NPG (Agarwal et al., 2021a), and PC-PG (Agarwal et al., 2020). While in this work, we use the simplest and most intuitive definition of coverage – the density ratio-based definition in Eq. 10 – extension to more general ones such as transfer error (Agarwal et al., 2020; 2021a) or concentrability coefficients that incorporate function class (e.g., Song et al. (2023b)) is straightforward. We defer the proof of the above theorem and the detailed constants that we omitted in the O notation to Appendix E. We include an extension of the above analysis to general preference setting in Appendix F.

5 Experiments

The implementation of REBEL follows Algorithm 1 with $\mu = \pi_t$. We empirically assess REBEL's performance on both natural language generation and text-guided image generation. Additional experiment details are in Appendix G.

5.1 Natural Language Generation

Task. We use the *TL;DR* summarization dataset (Stiennon et al., 2020)[2] where x is a forum post from Reddit and y is a summary generated by the policy. The dataset comprises human reference summaries and preference data. Following prior work (Stiennon et al., 2020; Rafailov et al., 2023; Ahmadian et al., 2024), we train the DPO baseline on the preference dataset, while conducting online RL (PPO, RLOO, REBEL) on the human reference dataset. We include results with three different model sizes: 1.4B, 2.8B, and 6.9B based on the pre-trained models from Pythia (Biderman et al., 2023). Each model is trained with a supervised fine-tuned (SFT) model and/or a reward model (RM) of the same size.

Evaluation. We evaluate each method by its balance between reward model score and KL-divergence with the SFT policy, testing the effectiveness of the algorithm in optimizing the regularized RL objective. To evaluate the quality of the generation, we compute the winrate (Rafailov et al., 2023)

[2]Dataset available at `https://github.com/openai/summarize-from-feedback`

Model size	Algorithm	Winrate (↑)	RM Score (↑)	KL($\pi\|\|\pi_{ref}$) (↓)
1.4B	SFT	24.5%	-0.52	-
	DPO	43.8%	0.11	30.9
	PPO	51.6%	1.73	**29.1**
	REBEL	**55.3%**	**1.87**	32.4
2.8B	SFT	28.4%	-0.40	-
	DPO	53.5%	2.41	66.5
	PPO	67.2%	2.37	**27.4**
	REBEL	**70.3%**	**2.44**	29.2

Model size	Algorithm	Winrate
6.9B	SFT	44.6%
	DPO	68.2%
	REINFORCE	70.7%*
	PPO	77.6%‡
	RLOO ($k = 2$)	74.2%*
	RLOO ($k = 4$)	77.9%*
	REBEL	**78.0%**

Table 1: Results on *TL;DR* Summarization. The best-performing method for each size and metric is highlighted in bold and the second best is underlined. We note that REBEL outperforms all baselines here in terms of the winrate. Results with * are directly obtained from Ahmadian et al. (2024). Results with ‡ are directly obtained from Huang et al. (2024).

against human references using GPT4 (OpenAI, 2023). The winrate is computed from a randomly sampled subset (10%) of the test set with a total of 600 samples.

Method. We compare REBEL with baseline RL algorithms, PPO (Schulman et al., 2017), Direct Preference Optimization (DPO) (Rafailov et al., 2023), and REINFORCE (Williams, 1992) and its multi-sample extension, REINFORCE Leave-One-Out (RLOO) (Kool et al., 2019).

Quality Analysis. Table 1 presents a comparison between REBEL and with baseline methods. Notably, REBEL outperforms all the baselines on RM score with 1.4B and 2.8B parameters with a slightly larger KL than PPO. In addition, REBEL achieves the highest winrate under GPT4 when evaluated against human references, indicating the benefit of regressing the relative rewards. An ablation analysis on parameter η is in Appendix I and the trade-off between the reward model score and KL-divergence is in Appendix J.

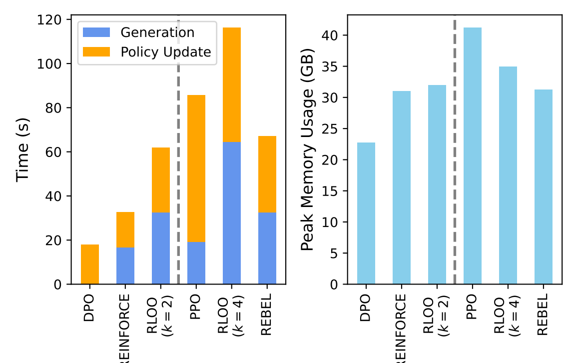


Figure 2: Plot of runtime and memory usage. Baselines on the left-hand side of the dashed line have lower winrates. Methods on the right-hand side of the dashed line have similar winrates to REBEL.

Runtime & Memory Analysis. We analyze the runtime and peak memory usage for 2.8B models using PPO, DPO, RLOO, and REBEL. The runtime includes both the generation time and the time required for policy updates. Both runtime and peak memory usage are measured on A6000 GPUs using the same hyperparameters detailed in Appendix G.2. The methods in the plots are arranged in ascending order based on winrates. To the right of the dashed line, PPO, RLOO ($k = 4$), and REBEL have the highest winrates, which are comparable among them.

While DPO and REINFORCE require less time and memory, their performance does not match up to REBEL, as shown in Table 1. RLOO ($k = 2$) has similar runtime and memory usage as REBEL since we set $\mu = \pi_t$, making REBEL also generate twice per prompt. However, RLOO ($k = 2$) has worse performance than REBEL. Compared to PPO and RLOO ($k = 4$), REBEL demonstrates shorter runtimes and lower peak memory usage. PPO is slow and requires more memory because it needs to update both two networks: policy network and value network. RLOO ($k = 4$) requires generating 4 responses per prompt which makes it slow and less memory efficient. Compared to the two baselines PPO and RLOO ($k = 4$) that achieve similar winrates as REBEL, we see that REBEL is more computationally tractable. REBEL is also noticeably simpler to implement than PPO since it does not learn value networks or compute the advantage estimation.

5.2 Image Generation

Task. We also consider the setting of image generation, where, given a consistency model (Song et al., 2023a) and a target reward function, we seek to train the consistency model to output images which garner a higher reward. We use 45 common animals as generation prompts similar to Black et al. (2023); Oertell et al. (2024)[3] and the latent consistency model (Luo et al., 2023) distillation of the Dreamshaper v7 model[4], a finetune of stable diffusion (Rombach et al., 2021).

Evaluation. We evaluate PPO and REBEL on its reward under the LAION aesthetic reward model for an equal number of reward queries/samples generated and an equal number of gradient updates. The aesthetic predictor is trained to predict human-labeled scores of images on a scale of 1 to 10. Images that tend to have the highest reward are artwork. Following Agarwal et al. (2021b), we report inter-quartile means (IQM) with 95% confidence intervals (CIs) across three seeds for both REBEL and PPO. The CIs were calculated with percentile bootstrap with stratified sampling over three random seeds.

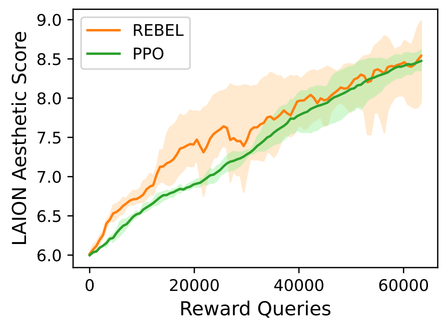


Figure 3: Learning curves as a function of reward queries to the LAION aesthetic predictor.

Method. We compare REBEL to a clipped, policy gradient objective (Black et al., 2023; Fan et al., 2024; Oertell et al., 2024) with the aim to optimize aesthetic quality to obtain high reward from the LAION aesthetic score predictor (Schuhmann, 2022). This baseline does not use critics or GAE for advantage estimates. However, the clipping objective is clearly motivated by PPO, and thus, we simply name this baseline as PPO in this section.

Quality Analysis. Figure 3 shows REBEL optimizes the consistency model faster during the beginning of training but eventually achieves similar performance to that of PPO. For our experiments, we tuned both batch size and learning rate for our algorithms, testing batch sizes of $[4, 8, 16]$ per gpu and learning rates $[1e-4, 3e-4, 6e-4, 1e-3]$. Note, the main difference in implementation between PPO and REBEL is the replacement of the clipped PPO objective with our regression objective. To maximize LAION-predicted aesthetic quality, both REBEL and PPO transform a model that produces plain images into one that produces artistic drawings. We found across multiple seeds that REBEL produced lush backgrounds when compared to PPO's generations. Please see Appendix H.2 for more examples of generated images.

6 Conclusion

We propose REBEL, an RL algorithm that reduces the problem of RL to solving a sequence of relative reward regression problems on iteratively collected datasets. In contrast to policy gradient approaches that require additional networks and heuristics like clipping to ensure optimization stability, REBEL requires that we can drive down training error on a least squares problem. This makes it strikingly simple to implement and scale. In theory, REBEL matches the best guarantees we have for RL algorithms in the agnostic setting, while in practice, REBEL is able to match and sometimes outperform methods that are far more complex to implement or expensive to run across both language modeling and guided image generation tasks.

[3]Dataset available at `https://github.com/Owen-Oertell/rlcm`
[4]Huggingface model card: SimianLuo/LCM_Dreamshaper_v7

References

Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. Reinforcement learning: Theory and algorithms, 2019.

Alekh Agarwal, Mikael Henaff, Sham Kakade, and Wen Sun. Pc-pg: Policy cover directed exploration for provable policy gradient learning. *Advances in neural information processing systems*, 33: 13399–13412, 2020.

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021a.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021b.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms, 2024.

Anthropic. Introducing the next generation of claude, 2024. URL https://www.anthropic.com/news/claude-3-family.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences, 2023.

J Andrew Bagnell and Jeff Schneider. Covariant policy search. In *Proceedings of the 18th international joint conference on Artificial intelligence*, pp. 1019–1024, 2003.

James Bagnell, Sham M Kakade, Jeff Schneider, and Andrew Ng. Policy search by dynamic programming. *Advances in neural information processing systems*, 16, 2003.

Philip J. Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data, 2023.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Daniele Calandriello, Daniel Guo, Remi Munos, Mark Rowland, Yunhao Tang, Bernardo Avila Pires, Pierre Harvey Richemond, Charline Le Lan, Michal Valko, Tianqi Liu, et al. Human alignment of large language models through online preference optimisation. *arXiv preprint arXiv:2403.08635*, 2024.

Jonathan D. Chang, Kiante Brantley, Rajkumar Ramamurthy, Dipendra Misra, and Wen Sun. Learning to generate better than your llm, 2023.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, 2017.

Qiwen Cui and Simon S Du. Provably efficient offline multi-agent reinforcement learning via strategy-wise bonus. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 11739–11751. Curran Associates, Inc., 2022a. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/4cca5640267b416cef4f00630aef93a2-Paper-Conference.pdf.

Qiwen Cui and Simon S Du. When are offline two-player zero-sum markov games solvable? *Advances in Neural Information Processing Systems*, 35:25779–25791, 2022b.

Miroslav Dudík, Katja Hofmann, Robert E Schapire, Aleksandrs Slivkins, and Masrour Zoghi. Contextual dueling bandits. In *Conference on Learning Theory*, pp. 563–587. PMLR, 2015.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep policy gradients: A case study on ppo and trpo, 2020.

Kawin Ethayarajh, Winnie Xu, and Douwe Kiela. Better, cheaper, faster llm alignment with kto, 2023. URL https://contextual.ai/better-cheaper-faster-llm-alignment-with-kto/.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Peter C Fishburn. Probabilistic social choice based on simple voting comparisons. *The Review of Economic Studies*, 51(4):683–692, 1984.

Martin Gardner. Mathematical games, Dec 1970. URL https://www.scientificamerican.com/article/mathematical-games-1970-12/.

Peter D Grünwald and A Philip Dawid. Game theory, maximum entropy, minimum discrepancy and robust bayesian decision theory, 2004.

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. Direct language model alignment from online ai feedback. *arXiv preprint arXiv:2402.04792*, 2024.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters, 2019.

Chloe Ching-Yun Hsu, Celestine Mendler-Dünner, and Moritz Hardt. Revisiting design choices in proximal policy optimization, 2020.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Shengyi Huang, Michael Noukhovitch, Arian Hosseini, Kashif Rasul, Weixun Wang, and Lewis Tunstall. The n+ implementation details of rlhf with ppo: A case study on tl;dr summarization, 2024.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 267–274, 2002.

Sham M Kakade. A natural policy gradient. In T. Dietterich, S. Becker, and Z. Ghahramani (eds.), *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL https://proceedings.neurips.cc/paper_files/paper/2001/file/4b86abe48d358ecf194c56c69108433e-Paper.pdf.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on robot learning*, pp. 651–673. PMLR, 2018.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. In S. Solla, T. Leen, and K. Müller (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 reinforce samples, get a baseline for free! In *DeepRLStructPred@ICLR*, 2019. URL https://api.semanticscholar.org/CorpusID:198489118.

Gerald H Kramer. On a class of equilibrium conditions for majority rule. *Econometrica*, 41(2):285–97, 1973. URL https://EconPapers.repec.org/RePEc:ecm:emetrp:v:41:y:1973:i:2:p:285-97.

Germain Kreweras. Aggregation of preference orderings. In *Mathematics and Social Sciences I: Proceedings of the seminars of Menthon-Saint-Bernard, France (1–27 July 1960) and of Gösing, Austria (3–27 July 1962)*, pp. 73–79, 1965.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. Rewardbench: Evaluating reward models for language modeling, 2024.

John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. *Advances in neural information processing systems*, 20, 2007.

Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback, 2023.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference, 2023.

Kenneth O May. Intransitivity, utility, and the aggregation of preference patterns. *Econometrica: Journal of the Econometric Society*, pp. 1–13, 1954.

Meta. Introducing meta llama 3: The most capable openly available llm to date, 2024. URL https://ai.meta.com/blog/meta-llama-3/.

Rémi Munos. Error bounds for approximate policy iteration. In *ICML*, volume 3, pp. 560–567. Citeseer, 2003.

Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Andrea Michi, et al. Nash learning from human feedback. *arXiv preprint arXiv:2312.00886*, 2023.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback, 2022.

Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization, 1983.

Owen Oertell, Jonathan D Chang, Yiyi Zhang, Kianté Brantley, and Wen Sun. Rl for consistency models: Faster reward guided text-to-image generation. *arXiv preprint arXiv:2404.03673*, 2024.

OpenAI. Gpt-4 technical report, 2023.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning, 2019.

Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*, pp. 745–750, 2007.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2023.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241*, 2022.

Juntao Ren, Gokul Swamy, Zhiwei Steven Wu, J Andrew Bagnell, and Sanjiban Choudhury. Hybrid inverse reinforcement learning. *arXiv preprint arXiv:2402.08848*, 2024.

Lorenz Richter, Ayman Boustati, Nikolas Nüsken, Francisco Ruiz, and Omer Deniz Akyildiz. Vargrad: a low-variance gradient estimator for variational inference. *Advances in Neural Information Processing Systems*, 33:13481–13492, 2020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacroce, Ahmed Awadallah, and Tengyang Xie. Direct nash optimization: Teaching language models to self-improve with general preferences. *arXiv preprint arXiv:2404.03715*, 2024.

Chrisoph Schuhmann. Laion aesthetics. https://laion.ai/blog/laion-aesthetics/, 2022.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015a.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

Lior Shani, Yonathan Efroni, and Shie Mannor. Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5668–5675, 2020.

Paul B. Simpson. On Defining Areas of Voter Choice: Professor Tullock on Stable Voting. *The Quarterly Journal of Economics*, 83(3):478–490, 08 1969. ISSN 0033-5533. doi: 10.2307/1880533. URL https://doi.org/10.2307/1880533.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023a.

Yuda Song, Yifei Zhou, Ayush Sekhari, J. Andrew Bagnell, Akshay Krishnamurthy, and Wen Sun. Hybrid rl: Using both offline and online data can make rl efficient, 2023b.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2022.

Gokul Swamy, Sanjiban Choudhury, J Andrew Bagnell, and Steven Wu. Of moments and matching: A game-theoretic framework for closing the imitation gap. In *International Conference on Machine Learning*, pp. 10022–10032. PMLR, 2021.

Gokul Swamy, Christoph Dann, Rahul Kidambi, Zhiwei Steven Wu, and Alekh Agarwal. A minimaximalist approach to reinforcement learning from human feedback. *arXiv preprint arXiv:2401.04056*, 2024.

Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. Preference fine-tuning of llms should leverage suboptimal, on-policy data, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

Amos Tversky. Intransitivity of preferences. *Psychological review*, 76(1):31, 1969.

Yuanhao Wang, Qinghua Liu, and Chi Jin. Is RLHF more difficult than standard RL? a theoretical perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=sxZLrBqg50.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256, may 1992. ISSN 0885-6125. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Wei Xiong, Han Zhong, Chengshuai Shi, Cong Shen, Liwei Wang, and Tong Zhang. Nearly minimax optimal offline reinforcement learning with linear function approximation: Single-agent MDP and markov game. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=UP_GHHPw7rP.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint, 2024.

Chenlu Ye, Wei Xiong, Yuheng Zhang, Nan Jiang, and Tong Zhang. A theoretical analysis of nash learning from human feedback under general kl-regularized preference. *arXiv preprint arXiv:2402.07314*, 2024.

Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

Han Zhong, Wei Xiong, Jiyuan Tan, Liwei Wang, Tong Zhang, Zhaoran Wang, and Zhuoran Yang. Pessimistic minimax value iteration: Provably efficient equilibrium learning from offline datasets. In *International Conference on Machine Learning*, pp. 27117–27142. PMLR, 2022.

Yifei Zhou, Ayush Sekhari, Yuda Song, and Wen Sun. Offline data enhanced on-policy policy gradient with provable guarantees. *arXiv preprint arXiv:2311.08384*, 2023.

Banghua Zhu, Michael Jordan, and Jiantao Jiao. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In *International Conference on Machine Learning*, pp. 43037–43067. PMLR, 2023.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2020.

A Related Work

Policy Gradients. Policy gradient (PG) methods (Nemirovskij & Yudin, 1983; Williams, 1992; Konda & Tsitsiklis, 1999; Kakade, 2001; Schulman et al., 2017) are a prominent class of RL algorithms due to their direct, gradient-based policy optimization, robustness to model mis-specification (Agarwal et al., 2020), and scalability to modern AI applications from fine-tuning LLMs (Stiennon et al., 2022) to optimizing text-to-image generators (Oertell et al., 2024).

Broadly speaking, we can taxonomize PG methods into two families. The first family is based on REINFORCE (Williams, 1992) and often includes variance reduction techniques (Kool et al., 2019; Richter et al., 2020; Zhu et al., 2023). While prior work by Ahmadian et al. (2024) has shown that REINFORCE-based approaches can outperform more complex RL algorithms like PPO on LLM fine-tuning tasks like *TL;DR*, we find that a properly optimized version of PPO still out-performs a REINFORCE baseline. The second family is *adaptive* PG techniques that *precondition* the policy gradient (usually with the inverse of the Fisher Information Matrix) to ensure it is *covariant* to re-parameterizations of the policy, which include NPG (Kakade, 2001; Bagnell & Schneider, 2003) and its practical approximations like TRPO (Schulman et al., 2015a) and PPO (Schulman et al., 2017). Intuitively, the preconditioning ensures that we make small changes in terms of action distributions, rather than in terms of the actual policy parameters, leading to faster and more stable convergence. Unfortunately, computing and then inverting the Fisher Information Matrix is computationally intensive and therefore we often resort to approximations in practice, as done in TRPO. However, these approximations are still difficult to apply to large-scale generative models, necessitating even coarser approximations like PPO. In contrast, `REBEL` does not need any such approximations to be implemented at scale, giving us a much closer connection between theory and practice.

Reward Regression. The heart of `REBEL` is a novel reduction from RL to iterative squared loss regression. While using regression to fit either the reward (Peters & Schaal, 2007) or the value (Peng et al., 2019) targets which are then used to extract a policy have previously been explored, our method instead takes a page from DPO (Rafailov et al., 2023) to implicitly parameterize the reward regressor in terms of the policy. This collapses the two stage procedure of prior methods into a single regression step.

Preference Fine-Tuning (PFT) of Generative Models. RL has attracted renewed interest due to its central role in "aligning" language models – i.e., adapting their distribution of prompt completions towards the set of responses preferred by human raters.

One family of techniques for PFT, often referred to as Reinforcement Learning from Human Feedback (RLHF) involves first fitting a reward model (i.e. a classifier) to the human preference data and then using this model to provide reward values to a downstream RL algorithm (often PPO) (Christiano et al., 2017; Ziegler et al., 2020). LLMs fine-tuned by this procedure include GPT-N (OpenAI, 2023), Claude-N (Anthropic, 2024), and Llama-N (Meta, 2024). Similar approaches have proved beneficial for tasks like summarization (Stiennon et al., 2022), question answering (Nakano et al., 2022), text-to-image generation (Lee et al., 2023), and instruction following (Ouyang et al., 2022).

Another family of techniques for PFT essentially treats the problem as supervised learning and uses a variety of ranking loss functions. It includes DPO (Rafailov et al., 2023), IPO (Azar et al., 2023), and KTO (Ethayarajh et al., 2023). These techniques are simpler to implement as they remove components like an explicit reward model, value network, and on-policy training from the standard RLHF setup. However, recent work finds their performance to be lesser than that of on-policy methods (Lambert et al., 2024; Tajwar et al., 2024), which agrees with our findings. This is perhaps caused by their lack of interaction during training, leading to the well-known covariate shift/compounding error issue (Ross et al., 2011; Swamy et al., 2021) and the associated lower levels of performance.

The third family of PFT techniques combines elements from the previous two: it involves running an offline algorithm *iteratively*, collecting on-policy preference feedback from either a supervisor model (Rosset et al., 2024; Xiong et al., 2024; Guo et al., 2024) or from a preference model fit on human data

(Calandriello et al., 2024). All of these approaches can be considered instantiations of the general SPO reduction proposed by Swamy et al. (2024), which itself can be thought of as a preference-based variant of DAgger (Ross et al., 2011). Recent work by Tajwar et al. (2024) confirms the empirical strength of these techniques. Our approach fits best into this family of techniques – we also iteratively update our model by solving a sequence of supervised learning problems over on-policy datasets. However, `REBEL` comes with several key differentiating factors from the prior work. First, we can run `REBEL` with datasets consisting of a mixture of on-policy and off-policy data with strong guarantees, enabling *hybrid training*, as previously explored in the RL (Song et al., 2023b; Ball et al., 2023; Zhou et al., 2023) and inverse RL (Ren et al., 2024) literature. Second, unlike all of the aforementioned works that regularize to the initial policy π_0 during updates, we perform *conservative* updates by regularizing π_{t+1} to π_t. Thus, for the prior work, it is difficult to prove convergence or monotonic improvement as the current policy can just bounce around a ball centered at π_0, a well-known issue in the theory of approximate policy iteration (Kakade & Langford, 2002; Munos, 2003). In contrast, by incorporating the prior policy's probabilities into our regression problem, we are able to prove stronger guarantees for `REBEL`.

B Proof of Claim 1

We prove claim 1 in this section. We start from deriving the Fisher information matrix.

$$
\begin{aligned}
F_t &:= \frac{1}{\eta^2}\mathbb{E}_{x,y\sim\pi_t,y'\sim\mu}\left(\nabla_\theta \ln \pi_{\theta_t}(y|x) - \nabla_\theta \ln \pi_{\theta_t}(y'|x)\right)\left(\nabla_\theta \ln \pi_{\theta_t}(y|x) - \nabla_\theta \ln \pi_{\theta_t}(y'|x)\right)^\top \\
&= \frac{2}{\eta^2}\mathbb{E}_{x,y\sim\pi_{mix}}\nabla_\theta \ln \pi_{\theta_t}(y|x)\nabla_\theta \ln \pi_{\theta_t}(y|x)^\top
\end{aligned}
$$

where the last equality uses the fact that cross terms from completing the square are zero. Now recall Eq. 9 which is a ordinarly least square regression problem. The minimum norm solution of the least square regression problem is:

$$
\begin{aligned}
\delta &= (\eta/2)\tilde{F}_t^\dagger\left(\mathbb{E}_{x,y\sim\pi_t,y'\sim\mu}\left(\nabla_\theta \ln \pi_{\theta_t}(y|x) - \nabla_\theta \ln \pi_{\theta_t}(y'|x)\right)\left(r(x,y) - r(x,y')\right)\right) \\
&= (\eta/2)\tilde{F}_t^\dagger\Big(\mathbb{E}_{x,y\sim\pi_t}\left[\nabla_\theta \ln \pi_{\theta_t}(y|x) r(x,y)\right] + \mathbb{E}_{x,y'\sim\mu}\left[\nabla_\theta \ln \pi_{\theta_t}(y'|x) r(x,y')\right] \\
&\qquad - \mathbb{E}_{x,y\sim\pi_t y'\sim\mu}\nabla_\theta \ln \pi_{\theta_t}(y'|x) r(x,y)\Big) \\
&= (\eta/2)\tilde{F}_t^\dagger\Big(\mathbb{E}_{x,y\sim\pi_t}\left[\nabla_\theta \ln \pi_{\theta_t}(y|x)[r(x,y) - \mathbb{E}_{y'\sim\pi_t(\cdot|x)}r(x,y')\right] \\
&\qquad\qquad + \mathbb{E}_{x,y\sim\mu}\left[\nabla_\theta \ln \pi_{\theta_t}(y|x)[r(x,y) - \mathbb{E}_{y'\sim\pi_t(\cdot|x)}r(x,y')\right]\Big) \\
&= (\eta)\tilde{F}_t^\dagger\left(\mathbb{E}_{x,y\sim(\pi_t+\mu)/2}\left[\nabla_\theta \ln \pi_{\theta_t}(y|x)[A^{\pi_t}(x,y)\right]\right)
\end{aligned}
$$

where we again use the fact that $\mathbb{E}_{y\sim\pi_{\theta_t}(\cdot|x)}\nabla_\theta \ln \pi_{\theta_t}(y|x) g(x) = 0$ for any function $g(x)$, and we define *Advantage* $A^\pi(x,y) := r(x,y) - \mathbb{E}_{y'\sim\pi(\cdot|x)}r(x,y')$.

C Extending REBEL to General Preferences

In the above discussion, we assume we are given access to a ground-truth reward function. However, in the generative model fine-tuning applications of RL, we often need to learn from human *preferences*, rather than rewards. This shift introduces a complication: not all preferences can be rationalized by an underlying utility function. In particular, *intransitive* preferences which are well-known to result from aggregation of different sub-populations or users evaluating different pairs of items on the basis of different features (May, 1954; Tversky, 1969; Gardner, 1970) cannot be accurately captured by a single reward model. To see this, note that if we have $a \succ b$, $b \succ c$, and $c \succ a$, it is impossible to have a reward model that simultaneously sets $\hat{r}(a) > \hat{r}(b)$, $\hat{r}(b) > \hat{r}(c)$, and $\hat{r}(c) > \hat{r}(a)$. As we increase the space of possible choices to that of all possible prompt completions, the probability of such intransitivities sharply increases (Dudík et al., 2015), as reflected in the high levels of annotator disagreement in LLM fine-tuning datasets (Touvron et al., 2023). Thus, rather than assuming access to a reward model, in such settings, we assume access to a *preference model* (Munos et al., 2023; Swamy et al., 2024; Rosset et al., 2024; Ye et al., 2024).

C.1 A Game-Theoretic Perspective on Learning from Preferences

More specifically, for any tuple (x, y, y'), we assume we have access to $\mathcal{P}(y \succ y'|x)$: the probability that y is preferred to y'. We then define our preference model l as

$$l(x, y, y') \triangleq 2 \cdot \mathcal{P}(y \succ y'|x) - 1. \tag{11}$$

Observe that $l(x, y, y') \in [-1, 1]$ is skew-symmetric, i.e., $l(x, y, y) = 0$, $l(x, y, y') + l(x, y', y) = 0$ for all $x \in \mathcal{X}, y, y' \in \mathcal{Y}$. If the learner can only receive a binary feedback $o \in \{0, 1\}$ indicating the preference between y and y', we assume o is sampled from a Bernoulli distribution with mean $\mathcal{P}(y \succ y'|x)$, where $o = 1$ means that y is preferred over y' and 0 otherwise.

Given access to such a preference model, a solution concept to the preference aggregation problem with deep roots in the social choice theory literature (Kreweras, 1965; Fishburn, 1984; Kramer, 1973; Simpson, 1969) and the dueling bandit literature (Yue et al., 2012; Dudík et al., 2015) is that of a minimax winner (MW) π_{MW}: the Nash Equilibrium strategy of the symmetric two-player zero-sum game with l as a payoff function. In particular, due to the skew-symmetric property of l, Swamy et al. (2024) proved that there exists a policy π_{MW} such that

$$\max_{\pi} \mathbb{E}_{x\sim\rho, y\sim\pi(\cdot|x), y'\sim\pi_{\mathsf{MW}}(\cdot|x)} \left[l(x, y, y')\right] = \min_{\pi} \mathbb{E}_{x\sim\rho, y\sim\pi_{\mathsf{MW}}(\cdot|x), y'\sim\pi(\cdot|x)} \left[l(x, y, y')\right].$$

This implies that $(\pi_{\mathsf{MW}}, \pi_{\mathsf{MW}})$ is a Nash Equilibrium (Wang et al., 2023; Munos et al., 2023; Swamy et al., 2024; Ye et al., 2024). As is standard in game solving, our objective is to obtain an ϵ-approximate MW $\widehat{\pi}$ measured by the duality gap (DG):

$$\mathsf{DG}(\widehat{\pi}) := \max_{\pi} \mathbb{E}_{x\sim\rho, y\sim\pi(\cdot|x), y'\sim\widehat{\pi}(\cdot|x)} \left[l(x, y, y')\right] - \min_{\pi} \mathbb{E}_{x\sim\rho, y\sim\widehat{\pi}(\cdot|x), y'\sim\pi(\cdot|x)} \left[l(x, y, y')\right] \leq \epsilon.$$

In the following discussion, we will use $l(x, y, \pi)$ to denote $\mathbb{E}_{y'\sim\pi(\cdot|x)}[l(x, y, y')]$ and $l(\pi, \pi')$ to denote $\mathbb{E}_{x\sim\rho, y\sim\pi(\cdot|x), y'\sim\pi'(\cdot|x)}[l(x, y, y')]$ for notational convenience.

C.2 Self-Play Preference Optimization (SPO) with REBEL as Base Learner

We can straightforwardly extend REBEL to the general preference setting via an instantiation of the Self-Play Preference Optimization (SPO) reduction of Swamy et al. (2024). In short, Swamy et al. (2024) prove that rather than performing adversarial training, we are able to perform a simple and stable *self-play* procedure while retaining strong theoretical guarantees. Practically, this corresponds to sampling at leas two completions from the current policy, querying a learned preference / supervisor model on each pair, and using the win rate for each completion as its reward. We will now describe how we can adapt REBEL to this mode of feedback.

Assuming that we can query the preference oracle $l(x, y, y')$ at will, we can modify the least square objective Eq. (1) to

$$\theta_{t+1} := \underset{\theta}{\operatorname{argmin}} \sum_{x,y,y',y''\in\mathcal{D}_t} \left(\frac{1}{\eta}\left(\ln\frac{\pi_\theta(y|x)}{\pi_{\theta_t}(y|x)} - \ln\frac{\pi_\theta(y'|x)}{\pi_{\theta_t}(y'|x)}\right) - \left(l(x,y,y'') - l(x,y',y'')\right)\right)^2$$

where $x \sim \rho, y \sim \pi_t(\cdot|x), y'' \sim \pi_t(\cdot|x), y' \sim \mu(\cdot|x)$. When the exact value of $l(x, y, y')$ is unavailable but only a binary preference feedback $o_{y,y'} \in \{0, 1\}$ sampling from Bernoulli with mean $l(x, y, y')$ is available, we can just replace $l(x, y, y'') - l(x, y', y'')$ by $o_{y,y'} - o_{y',y''}$. It is easy to see that the Bayes optimal of the above least square regression problem is equal to:

$$\mathbb{E}_{y''\sim\pi_t(\cdot|x)} l(x,y,y'') - \mathbb{E}_{y''\sim\pi_t(\cdot|x)} l(x,y',y'') = l(x,y,\pi_t) - l(x,y',\pi_t).$$

Swamy et al. (2024) define an iteration-dependent reward $r_t(x, y) := \mathbb{E}_{y''\sim\pi_t(\cdot|x)} l(x, y, y'') = l(x, y, \pi_t)$. Thus, the above regression problem can be understood as an extension of `REBEL` to the setting where the reward function changes at each iteration t. Swamy et al. (2024) shows that running the exact MD (Eq. 3) with this iteration-dependent reward function r_t leads to fast convergence to an approximate Minimax Winner, a property that we will use to provide the regret bound of `REBEL` in the general preference setting while accounting for nonzero mean squared error.

D Justification for Assumption 1

Intuitively, this assumption is saying that there is a function in our class of regressors that is able to accurately fit the difference of rewards. Recall that our class of regressors is isomorphic to our policy class. Therefore, as long as our class of policies is expressive, we would expect this assumption to hold with small ϵ. For all domains we consider, our policy class is a flexible set of generative models (e.g. Transformer-based LLMs or diffusion models). Thus, we believe it is reasonable to believe this assumption holds in practice – see Figure 6 in Appendix K for empirical evidence of this point and Example 1 for more discussion.

More formally, the above assumption bounds the standard **in-distribution generalization error** (v.s. the point-wise guarantee in Eq. 8) of a well-defined supervised learning problem: least squares regression. The generalization error ϵ captures the possible errors from the learning process for θ_{t+1} and it could depend on the complexity of the policy class and the number of samples used in the dataset $\mathcal{D}_t$. For instance, when the the function $\ln \pi - \ln \pi'$ induced by the log-difference of two policies (π, π') are rich enough (e.g., policies are deep neural networks) to capture the reward difference, then ϵ in this assumption converges to zero as we increase the number of training data. Note that while ϵ can be small, it does *not* imply that the learned predictor will have a small prediction error in a point-wise manner – it almost certainly will not.

Example 1. *One simple example is when $\pi(y|x) \propto \exp(\theta^\top \phi(x,y))$ for some features $\phi(x,y)$. In this case, $\ln(\pi(y|x)/\pi_t(y|x)) - \ln(\pi(y'|x)/\pi_t(y'|x)) = (\theta - \theta_t)^\top (\phi(x,y) - \phi(x,y'))$, which means that our regression problem in Eq. 1 is a classic linear regression problem. When the reward $r(x,y)$ is also linear in feature $\phi(x,y)$, then Eq. 1 is a well-specified linear regression problem, and ϵ typically scales in the rate of $O\left(d/|\mathcal{D}_t|\right)$ with d being the dimension of feature ϕ.*

We can extend the above example to the case where ϕ is the feature corresponding to some kernel, e.g., RBF kernel or even Neural Tangent Kernel, which allows us to capture the case where π is a softmax wide neural network with the least square regression problem solved by gradient flow. The error ϵ again scales poly$(d/|\mathcal{D}_t|)$, where d is the effective dimension of the corresponding kernel.

E Proof of Theorem 1

In this section, we provide the proof of theorem 1. For notation simplicity, throughout the proof, we denote π_t for π_{θ_t}, and define $f_t(x,y) := \frac{1}{\eta}\ln\frac{\pi_{t+1}(y|x)}{\pi_t(y|x)}$.

The following lemma shows that the learned function f_t can predict reward r well under both π_t and μ up to terms that are y-independent.

Lemma 1. *Consider any $t \in [T]$. Define $\Delta(x,y) = f_t(x,y) - r(x,y)$. Define $\Delta_{\pi_t}(x) = \mathbb{E}_{y\sim\pi_t(\cdot|x)}\Delta(x,y)$ and $\Delta_\mu(x) = \mathbb{E}_{y\sim\mu(\cdot|x)}\Delta(x,y)$. Under assumption 1, for all t, we have the following:*

$$\mathbb{E}_{x,y\sim\pi_t(\cdot|x)}\left(f_t(x,y) - r(x,y) - \Delta_{\pi_t}(x)\right)^2 \le \epsilon, \tag{12}$$

$$\mathbb{E}_{x,y\sim\mu(\cdot|x)}\left(f_t(x,y) - r(x,y) - \Delta_\mu(x)\right)^2 \le \epsilon, \tag{13}$$

$$\mathbb{E}_x\left(\Delta_{\pi_t}(x) - \Delta_\mu(x)\right)^2 \le \epsilon. \tag{14}$$

Proof. From assumption 1, we have:

$$\begin{aligned}
&\mathbb{E}_{x,y_1\sim\pi_t,y_2\sim\mu}\left(f_t(x,y_1) - \Delta_{\pi_t}(x) - r(x,y_1) - (f_t(x,y_2) - \Delta_\mu(x) - r(x,y_2)) + \Delta_{\pi_t}(x) - \Delta_\mu(x)\right)^2 \\
&= \mathbb{E}_{x,y_1\sim\pi_t}\left(f_t(x,y_1) - \Delta_{\pi_t}(x) - r(x,y_1)\right)^2 + \mathbb{E}_{x,y_2\sim\mu}\left(f_t(x,y_2) - \Delta_\mu(x) - r(x,y_2)\right)^2 \\
&- 2\mathbb{E}_{x,y_1\sim\pi_t,y_2\sim\mu}\left(f_t(x,y_1) - \Delta_{\pi_t}(x) - r(x,y_1)\right)\left(f_t(x,y_2) - \Delta_\mu(x) - r(x,y_2)\right) \\
&+ 2\mathbb{E}_{x,y_1\sim\pi_t}\left(f_t(x,y_1) - \Delta_{\pi_t}(x) - r(x,y_1)\right)\left(\Delta_{\pi_t}(x) - \Delta_\mu(x)\right) \\
&- 2\mathbb{E}_{x,y_2\sim\pi_t}\left(f_t(x,y_2) - \Delta_\mu(x) - r(x,y_2)\right)\left(\Delta_{\pi_t}(x) - \Delta_\mu(x)\right) + \mathbb{E}_x(\Delta_1(x) - \Delta_2(x))^2 \\
&= \mathbb{E}_{x,y_1\sim\pi_t}\left(f_t(x,y_1) - \Delta_{\pi_t}(x) - r(x,y_1)\right)^2 + \mathbb{E}_{x,y_2\sim\mu}\left(f_t(x,y_2) - \Delta_\mu(x) - r(x,y_2)\right)^2 \\
&\qquad + \mathbb{E}_x(\Delta_{\pi_t}(x) - \Delta_\mu(x))^2 \le \epsilon.
\end{aligned}$$

In the above, we first complete the square, and then we only keep terms that are not necessarily zero. Since all the remaining three terms are non-negative, this concludes the proof. $\square$

By the definition of f_t, we have $\Delta(x,y) = \frac{1}{\eta}\ln\frac{\pi_{t+1}(y|x)}{\pi_t(y|x)} - r(x,y)$. Taking exp on both sides, we get:

$$\forall x,y: \ \pi_{t+1}(y|x) = \pi_t(y|x)\exp\left(\eta(r(x,y) + \Delta(x,y))\right) = \frac{\pi_t(y|x)\exp(\eta(r(x,y) + \Delta(x,y) - \Delta_\mu(x)))}{\exp(-\eta\Delta_\mu(x))}$$

Denote $g_t(x,y) := r(x,y) + \Delta(x,y) - \Delta_\mu(x)$, and the advantage $A_t(x,y) = g_t(x,y) - \mathbb{E}_{y'\sim\pi_t(\cdot|x)}g_t(x,y')$. We can rewrite the above update rule as:

$$\forall x,y: \ \pi_{t+1}(y|x) \propto \pi_t(y|x)\exp(\eta A_t(x,y)) \tag{15}$$

In other words, the algorithm can be understood as running MD on the sequence of A_t for $t = 0$ to $T-1$. The following lemma is the standard MD regret lemma.

Lemma 2. *Assume $\max_{x,y,t}|A_t(x,y)| \le A \in \mathbb{R}^+$, and $\pi_0(\cdot|x)$ is uniform over $\mathcal{Y}$. Then with $\eta = \sqrt{\ln(|\mathcal{Y}|)/(A^2T)}$, for the sequence of policies computed by* `REBEL`*, we have:*

$$\forall \pi, x: \ \sum_{t=0}^{T-1}\mathbb{E}_{y\sim\pi(\cdot|x)}A_t(x,y) \le 2A\sqrt{\ln(|\mathcal{Y}|)T}.$$

Proof. For completeness, we provide the proof here. Start with $\pi_{t+1}(y|x) = \pi_t(y|x)\exp(\eta A_t(x,y))/Z_t(x)$ where $Z_t(x)$ is the normalization constant, taking log on both sides, and add $\mathbb{E}_{y\sim\pi(\cdot|x)}$, we have:

$$-\mathrm{KL}(\pi(\cdot|x)||\pi_{t+1}(\cdot|x)) = -\mathrm{KL}(\pi(\cdot|x)||\pi_t(\cdot|x)) + \eta\mathbb{E}_{y\sim\pi(\cdot|x)}A_t(x,y) - \mathbb{E}_{y\sim\pi(\cdot|x)}\ln Z_t(x).$$

Rearrange terms, we get:

$$-\mathrm{KL}(\pi(\cdot|x)||\pi_t(\cdot|x)) + \mathrm{KL}(\pi(\cdot|x)||\pi_{t+1}(\cdot|x) = \mathbb{E}_{y\sim\pi(\cdot|x)}\left[-\eta A_t(x,y) + \ln Z_t(x)\right]$$

For $\ln Z_t(x)$, using the condition that $\eta \leq 1/A$, we have $\eta A_t(x,y) \leq 1$, which allows us to use the inequality $\exp(x) \leq 1 + x + x^2$ for any $x \leq 1$, which lead to the following inequality:

$$\begin{aligned}
\ln Z_t(x) &= \ln\left(\mathbb{E}_{y\sim\pi(\cdot|x)} \exp(\eta A_t(x,y))\right) \\
&\leq \ln\left(\sum_y \pi_t(y|x)\left(1 + \eta A_t(x,y) + \eta^2 A_t(x,y)^2\right)\right) \\
&\leq \ln\left(1 + 0 + \eta^2 A^2\right) \leq \eta^2 A^2,
\end{aligned}$$

where the last inequality uses $\ln(1+x) \leq x$, and we used the fact that $\mathbb{E}_{y\sim\pi_t(x)} A_t(x,y) = 0$ due to the definition of advantage A_t. Thus, we have:

$$-\mathrm{KL}(\pi(\cdot|x)||\pi_t(\cdot|x)) + \mathrm{KL}(\pi(\cdot|x)||\pi_{t+1}(\cdot|x) \leq -\mathbb{E}_{y\sim\pi(\cdot|x)}[A_t(x,y)] + \eta^2 A^2.$$

Sum over all iterations and do the telescoping sum, we get:

$$\sum_{t=0}^{T-1} \mathbb{E}_{y\sim\pi(\cdot|x)} A_t(x,y) \leq \mathrm{KL}(\pi(\cdot|x)||\pi_0(\cdot|x))/\eta + T\eta A^2 \leq \ln(|\mathcal{Y}|)/\eta + T\eta A^2.$$

With $\eta = \sqrt{\ln(|\mathcal{Y}|)/(A^2 T)}$, we conclude the proof. □

With the above, now we are ready to conclude the proof of the main theorem.

Proof of Theorem 1. Consider a comparator policy π^*. We start with the performance difference between π^* and the uniform mixture policy $\bar{\pi} := \sum_{t=0}^{T-1} \pi_t / T$:

$$\frac{1}{T}\sum_{t=0}^{T-1}\left(\mathbb{E}_{x,y\sim\pi^*(\cdot|x)} r(x,y) - \mathbb{E}_{x,y\sim\pi_t(\cdot|x)} r(x,y)\right) = \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}_{x,y\sim\pi^*(\cdot|x)}\left(A^{\pi_t}(x,y)\right),$$

where we define the real advantage $A^{\pi_t}(x,y) := r(x,y) - \mathbb{E}_{y\sim\pi_t(\cdot|x)} r(x,y)$. Continue, we have:

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}_{x,y\sim\pi^*(\cdot|x)}\left(A^{\pi_t}(x,y)\right) \\
&= \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}_{x,y\sim\pi^*(\cdot|x)}\left(A_t(x,y)\right) + \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}_{x,y\sim\pi^*(\cdot|x)}\left(A^{\pi_t}(x,y) - A_t(x,y)\right) \\
&\leq 2A\sqrt{\frac{\ln(|\mathcal{Y}|)}{T}} + \frac{1}{T}\sum_{t=0}^{T-1}\sqrt{\mathbb{E}_x \mathbb{E}_{y\sim\pi^*(\cdot|x)}(A^{\pi_t}(x,y) - A_t(x,y))^2}
\end{aligned}$$

where the last inequality uses Lemma 2. We now just need to bound $\mathbb{E}_{y\sim\pi^*(\cdot|x)}(A^{\pi_t}(x,y) - A_t(x,y))^2$.

$$\begin{aligned}
\mathbb{E}_x \mathbb{E}_{y\sim\pi^*(\cdot|x)}(A^{\pi_t}(x,y) - A_t(x,y))^2 &= \mathbb{E}_x \mathbb{E}_{y\sim\mu(\cdot|x)} \frac{\pi^*(y|x)}{\mu(y|x)}(A^{\pi_t}(x,y) - A_t(x,y))^2 \\
&\leq C_{\pi^*} \mathbb{E}_{x,y\sim\mu(\cdot|x)}(A^{\pi_t}(x,y) - A_t(x,y))^2
\end{aligned}$$

where the last inequality uses the definition of concentrability coefficient C_{π^*}. We now bound $\mathbb{E}_{x,y\sim\mu(\cdot|x)}(A^{\pi_t}(x,y) - A_t(x,y))^2$. Recall the definiton of A_t from Lemma 2.

$$\begin{aligned}
&\mathbb{E}_{x,y\sim\mu(\cdot|x)}(A^{\pi_t}(x,y) - A_t(x,y))^2 \\
&= \mathbb{E}_{x,y\sim\mu(\cdot|x)}(r(x,y) - \mathbb{E}_{y'\sim\pi_t(\cdot|x)} r(x,y') - g_t(x,y) + \mathbb{E}_{y'\sim\pi_t(\cdot|x)} g_t(x,y'))^2 \\
&\leq 2\mathbb{E}_{x,y\sim\mu(\cdot|x)}\left(r(x,y) - g_t(x,y)\right)^2 + 2\mathbb{E}_x \mathbb{E}_{y'\sim\pi_t(\cdot|x)}\left(r(x,y') - g_t(x,y')\right)^2
\end{aligned}$$

Recall the $g_t(x,y) = r(x,y) + \Delta(x,y) - \Delta_\mu(x)$, and from Lemma 1, we can see that

$$\mathbb{E}_{x,y\sim\mu(\cdot|x)}(r(x,y) - g_t(x,y))^2 = \mathbb{E}_{x,y\sim\mu(\cdot|x)}(\Delta(x,y) - \Delta_\mu(x))^2 \le \epsilon.$$

For $\mathbb{E}_x\mathbb{E}_{y'\sim\pi_t(\cdot|x)}\left(r(x,y') - g_t(x,y')\right)^2$, we have:

$$\begin{aligned}
&\mathbb{E}_x\mathbb{E}_{y'\sim\pi_t(\cdot|x)}\left(r(x,y') - g_t(x,y')\right)^2 = \mathbb{E}_x\mathbb{E}_{y'\sim\pi_t(\cdot|x)}\left(\Delta(x,y') - \Delta_\mu(x)\right)^2 \\
&\quad = \mathbb{E}_x\mathbb{E}_{y'\sim\pi_t(\cdot|x)}\left(\Delta(x,y') - \Delta_{\pi_t}(x) + \Delta_{\pi_t}(x) - \Delta_\mu(x)\right)^2 \\
&\quad \le 2\mathbb{E}_x\mathbb{E}_{y'\sim\pi_t(\cdot|x)}\left(\Delta(x,y') - \Delta_{\pi_t}(x)\right)^2 + 2\mathbb{E}_x\left(\Delta_{\pi_t}(x) - \Delta_\mu(x)\right)^2 \le 4\epsilon,
\end{aligned}$$

where the last inequality uses Lemma 1 again. Combine things together, we can conclude that:

$$\mathbb{E}_x\mathbb{E}_{y\sim\pi^*(\cdot|x)}(A^{\pi_t}(x,y) - A_t(x,y))^2 \le C_{\pi^*}(10\epsilon).$$

Finally, for the regret, we can conclude:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{x,y\sim\pi^*(\cdot|x)}\left(A^{\pi_t}(x,y)\right) \le 2A\sqrt{\frac{\ln|\mathcal{Y}|}{T}} + \frac{1}{T}\sum_t\sqrt{C_{\pi^*}10\epsilon} = 2A\sqrt{\frac{\ln|\mathcal{Y}|}{T}} + \sqrt{C_{\pi^*}10\epsilon}.$$

$\square$

F Extension of analysis to General Preferences

Extending the above analysis to the general preference case is straightforward except that it requires a stronger coverage condition. This is because we want to find a Nash Equilibrium, which requires a comparison between the learned policy against all the other policies. Results from the Markov Game literature (Cui & Du, 2022b; Zhong et al., 2022; Cui & Du, 2022a; Xiong et al., 2023) and Cui & Du (2022b) have shown that the standard single policy coverage condition used in single-player optimization is provably not sufficient. In particular, they propose using a notion of *unilateral concentrability* for efficient learning, which can be defined as

$$C_{\mathsf{uni},\mu} := \max_{\pi,x,y,y''} \frac{\pi_{\mathsf{MW}}(y|x)\pi(y''|x)}{\mu(y|x)\mu(y''|x)},$$

in the general preference setting. Notably, the above unilateral concentrability coefficient $C_{\mathsf{uni},\mu}$ is equivalent to $C_\mu := \max_{\pi,x,y} \frac{\pi(y|x)}{\mu(y|x)}$ since $C_\mu \leq C_{\mathsf{uni},\mu} \leq C_\mu^2$. Therefore in the following discussion, we will use C_μ as the coverage condition. In addition, we also assume the generalization error of the regression problem is small,

Assumption 2 (Regression generalization bounds for general preference)**.** *Over T iterations, assume that for all t, we have:*

$$\mathbb{E}_{x\sim\rho, y\sim\pi_t(\cdot|x), y'\sim\mu(\cdot|x)} \left(\frac{1}{\eta}\left(\ln \frac{\pi_{\theta_{t+1}}(y|x)}{\pi_{\theta_t}(y|x)} - \ln \frac{\pi_{\theta_{t+1}}(y'|x)}{\pi_{\theta_t}(y'|x)} \right) - (l(x,y,\pi_t) - l(x,y',\pi_t)) \right)^2 \leq \epsilon,$$

for some ϵ.

Under the above coverage condition and generalization bound, we can show that `REBEL` is able to learn an approximate Minimax Winner:

Theorem 2. *With assumption 2, after T many iterations, with a proper learning rate η, the policy $\widehat{\pi} = \mathsf{Unif}(\{\pi_t\}_{t=1}^T)$ satisfies that:*

$$\mathsf{DG}(\widehat{\pi}) \leq O\left(\sqrt{\frac{1}{T}} + \sqrt{C_\mu \epsilon.} \right).$$

Here the O-notation hides problem-dependent constants that are independent of ϵ, C_μ, T.

Note that the coverage condition here is much stronger than the single policy coverage condition in the RL setting. We conjecture that this is the cost one has to pay by moving to the more general preference setting and leaving the investigation of the necessarily coverage condition for future work.

F.1 Proof of Theorem 2

Recall that $r_t(x,y) = l(x,y,\pi_t)$. Let us define $\Delta^t(x,y) := f_t(x,y) - r_t(x,y)$, $\Delta^t_{\pi_t}(x) := \mathbb{E}_{y\sim\pi_t(\cdot|x)}\Delta^t(x,y)$ and $\Delta^t_\mu(x) := \mathbb{E}_{y\sim\mu(\cdot|x)}\Delta^t(x,y)$. Then following the same arguments in Lemma 1, we have

$$\mathbb{E}_{x\sim\rho, y\sim\pi_t(\cdot|x)} \left[\left(f_t(x,y) - r_t(x,y) - \Delta^t_{\pi_t}(x) \right)^2 \right] \leq \epsilon, \tag{16}$$

$$\mathbb{E}_{x\sim\rho, y\sim\mu(\cdot|x)} \left[\left(f_t(x,y) - r_t(x,y) - \Delta^t_{\mu}(x) \right)^2 \right] \leq \epsilon, \tag{17}$$

$$\mathbb{E}_{x\sim\rho} \left[\left(\Delta^t_{\pi_t}(x) - \Delta^t_\mu(x) \right)^2 \right] \leq \epsilon. \tag{18}$$

With slight abuse of the notation, We also use g_t and $A_t(x,y)$ to denote $r_t(x,y)+\Delta^t(x,y)-\Delta^t_\mu(x,y)$ and $g_t(x,y)-\mathbb{E}_{y'\sim\pi_t(\cdot|x)}g_t(x,y')$. Then following the same arguments in Lemma 2,

$$\forall \pi, x: \sum_{t=0}^{T-1} \mathbb{E}_{y\sim\pi(\cdot|x)} A_t(x,y) \le 2A\sqrt{\ln(|\mathcal{Y}|)T}. \tag{19}$$

Note that we have

$$\begin{aligned}
\max_\pi l(\pi,\widehat{\pi}) &= \max_\pi \frac{1}{T}\sum_{t=1}^{T} l(\pi,\pi_t)\\
&= \max_\pi \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{x\sim\rho,y\sim\pi(\cdot|x)}[r_t(x,y)] = \max_\pi \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{x\sim\rho,y\sim\pi(\cdot|x)}[A^{t,\pi_t}(x,y)],
\end{aligned}$$

where $A^{t,\pi_t} := r_t(x,y)-\mathbb{E}_{y\sim\pi_t(\cdot|x)}[r_t(x,y)]$. The last step is due to the skew symmetry of l, i.e., $\mathbb{E}_{y\sim\pi_t(\cdot|x)}[r_t(x,y)] = l(x,\pi_t,\pi_t) = 0$. Then by following the same arguments in the proof of Theorem 1, with (16)(17)(18)(19), we have for any policy π,

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{x\sim\rho,y\sim\pi(\cdot|x)}\left(A^{t,\pi_t}(x,y)\right) \le 2A\sqrt{\frac{\ln|\mathcal{Y}|}{T}} + \sqrt{10C_{\mu\to\pi}\epsilon}.$$

This implies that

$$\max_\pi l(\pi,\widehat{\pi}) \le \max_\pi\left(2A\sqrt{\frac{\ln|\mathcal{Y}|}{T}} + \sqrt{10C_{\mu\to\pi}\epsilon}\right) \le 2A\sqrt{\frac{\ln|\mathcal{Y}|}{T}} + \sqrt{10C_\mu\epsilon}.$$

Note that due to the skew symmetry of l, we have

$$\begin{aligned}
\min_\pi l(\widehat{\pi},\pi) &= \min_\pi \mathbb{E}_{x\sim\rho,y\sim\widehat{\pi}(\cdot|x),y'\sim\pi(\cdot|x)}\left[l(x,y,y')\right] = -\max_\pi \mathbb{E}_{x\sim\rho,y\sim\widehat{\pi}(\cdot|x),y'\sim\pi(\cdot|x)}\left[-l(x,y,y')\right]\\
&= -\max_\pi \mathbb{E}_{x\sim\rho,y\sim\pi(\cdot|x),y'\sim\widehat{\pi}(\cdot|x)}\left[l(x,y,y')\right] = -\max_\pi l(\pi,\widehat{\pi}) \ge -2A\sqrt{\frac{\ln|\mathcal{Y}|}{T}} - \sqrt{10C_\mu\epsilon}.
\end{aligned}$$

Therefore we have

$$\mathsf{DG}(\widehat{\pi}) \le 4A\sqrt{\frac{\ln|\mathcal{Y}|}{T}} + 2\sqrt{10C_\mu\epsilon}.$$

G Additional Experiment Details

G.1 Dataset Details

We present dataset details in Table 2.

Table 2: Dataset split, prompts, and maximum generation length for *TL;DR* summarization

Dataset	Train/Val/Test	Prompt	Generation Length
Human Reference	117K/6.45K/6.55K	"TL;DR:"	53
Preference	92.9K/83.8K/-	"TL;DR:"	53

G.2 Hyperparameter Details

Parameter setting for *TL;DR* summarization

Setting	**Parameters**	
SFT & RM	batch size: 64 learning rate: 3e-6	schedule: linear decay train epochs: 1
PPO	batch size: 512 learning rate: 3e-6 schedule: linear decay train epochs: 1 num epochs: 4	discount factor: 1 gae λ: 0.95 clip ratio: 0.2 value function coeff: 0.1 kl coefficient: 0.05
DPO	batch size: 64 learning rate: 3e-6 schedule: linear decay	train epochs: 1 β: 0.05
`REBEL`	batch size: 512 learning rate: 3e-6 schedule: linear decay train epochs: 1	num epochs: 4 η: 1.0 kl coefficient: 0.05
LoRA Adapter Config	r: 1024 α: 2048	dropout: 0.0 bias: False
Generation	sampling: true top k: 0.0 top p: 1.0	min length: 53 max new tokens: 53 temperature: 0.1

G.3 Model Details

G.3.1 TL;DR Summarization

For SFT models, we train a Pythia 1.4B (Biderman et al., 2023)[5] model for 1 epoch over the dataset with human references as labels, and use the existing fine-tuned 2.8B[6] and 6.9B[7] models. For reward models, we train a Pythia 1.4B parameter model for 1 epoch over the preference dataset and use the existing reward models with 2.8B[8] and 6.9B[9] parameters. For both `REBEL` and baseline methods using 1.4B and 2.8B parameters, we trained the policy and/or the critic using **low-rank adapters (LoRA)** (Hu et al., 2022) on top of our SFT and/or reward model respectively. For the 6.9B models, we perform **full-parameter** training.

[5]HuggingFace Model Card: EleutherAI/pythia-1.4b-deduped
[6]HuggingFace Model Card: vwxyzjn/EleutherAI_pythia-2.8b-deduped___sft___tldr
[7]HuggingFace Model Card: vwxyzjn/EleutherAI_pythia-6.9b-deduped___sft___tldr
[8]HuggingFace Model Card: vwxyzjn/EleutherAI_pythia-2.8b-deduped___reward___tldr
[9]HuggingFace Model Card: vwxyzjn/EleutherAI_pythia-6.9b-deduped___reward___tldr

G.4 Winrate Details

We are using `gpt-4-0613` checkpoint for winrate computations. Below we show the prompt for winrate evaluation and an example evaluation from GPT4.

Prompt for Winrate

Which of the following summaries does a better job of summarizing the most important points in the given forum post, without including unimportant or irrelevant details? Judge based on accuracy, coverage, and coherence.

Post:
{{post}}

Summary A:
{{summarya}}

Summary B:
{{summaryb}}

Instructions:
FIRST provide a one-sentence comparison of the two summaries, explaining which you prefer and why. SECOND, on a new line, state only "A" or "B" to indicate your choice.
Your response should use the format:
Comparison: <one-sentence comparison and explanation >
Preferred: <"A" or "B">

Example Evaluation from GPT4

Prompt	SUBREDDIT: r/AskReddit TITLE: How do you get someone out of your head? POST: Hi, I'm 22, and I have been with my girlfriend for 5 years now. We recently moved together. We've always loved each other intensely. Problem, I recently started to have feelings for an other person (a friend). This person has had a boyfriend for now 3 years, and has absolutely no ideas. Those feelings were so strong, it was hard to hide them. After 2 months of me being distant and really sad, my girlfriend forced me to say what was bothering me. I'm not a good liar, and now she knows. We decided to give us a week alone, I went to my parents. Now, I'm completely lost. I keep on thinking about this person, and I hate that. I would like for those feelings to go away, to leave me alone. But I can't. What do I do? It's been 3 months now, and I'm just desperate. TL;DR:
Reference (Summary A)	long relationship; fell in love with an other person; admitted it; would like it to disappear, though it doesn't.
`REBEL` **Generation (Summary B)**	I recently started to have feelings for an other person (a friend). We decided to give us a week alone, I keep on thinking about that person, and I hate it. What do I do?
Evaluation from GPT4	Comparison: Summary A is too brief and rather disjointed, while Summary B more accurately conveys the emotional conflict portrayed in the forum post in a coherent manner. Preferred: B

H Example Generations

H.1 Summarization Generations from a 2.8B model

Prompt	SUBREDDIT: r/relationships TITLE: Me [18 M] with my college friend [19 F] about 1 month, I wanna hang out with her but I don't have a car. POST: This is the link for an earlier thread I posted about her She basically doesn't have any friends and will spend all her spring break at home playing video games. I wanna hang out with her, but both of us don't have a car. I live with my grandma 10 minutes away from her house. Would it be weird to ask her to spend a day with us playing video games and eating pizza? Another thing I could do is ask her to go see a movie with me, but I would just meet her there since I'll be going with a bike. Do you guys think it's okay to do that? I also have 0 experience with girls. You can say this is the first girl in my life. TL;DR:
Generation	My college friend doesn't have any friends and spends all her spring break at home playing video games. I wanna hang out with her but neither of us have a car. Do you think it's okay to ask her to hang out?
Prompt	SUBREDDIT: r/AskReddit TITLE: Reddit, what kind of effect do you think that crime shows/movies might have in reducing the occurrence of thought-out, "intelligent" crime? POST: I had this thought a few days ago when watching an episode of NCIS and they did the usual "create a super high resolution image from a very low resolution image." I got to wondering how much of it is real and how much of it is fake (or, conversely, how much might be under-exaggerated). Based on these types of shows/movies (bank robbery, conning, etc.) where the bad guy nearly 100% of the time gets caught, only an idiot would think it would be a good idea to do any of this stuff, and from my [limited] knowledge of the general criminal, I'm under the impression that they are mainly uneducated people in dire situations (or psychopaths/mentally unstable.. but I'm thinking mostly with regards the heist type of crime – not assault or murder) that are driven to extreme lengths to make the "easy"/"quick" dollar. So I wondered, is it possible that these shows/movies that show the criminal, no matter how smart and creative, almost always getting caught could actually be (subconsciously or consciously) reducing the amount of more well thought through crime throughout the world? TL;DR:
Generation	I had this thought recently watching NCIS episode where criminals almost always get caught, is it possible that these crime shows/movies that show the criminal almost always getting caught could actually be (subconsciously or consciously) reducing the crime?

H.2 Image Generations

H.2.1 Comparison between PPO and REBEL

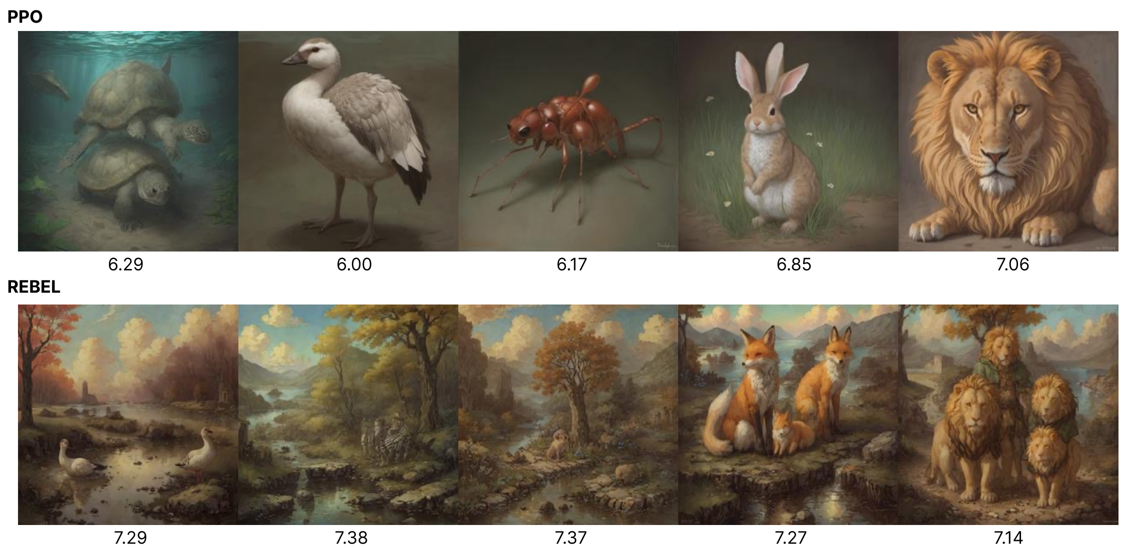
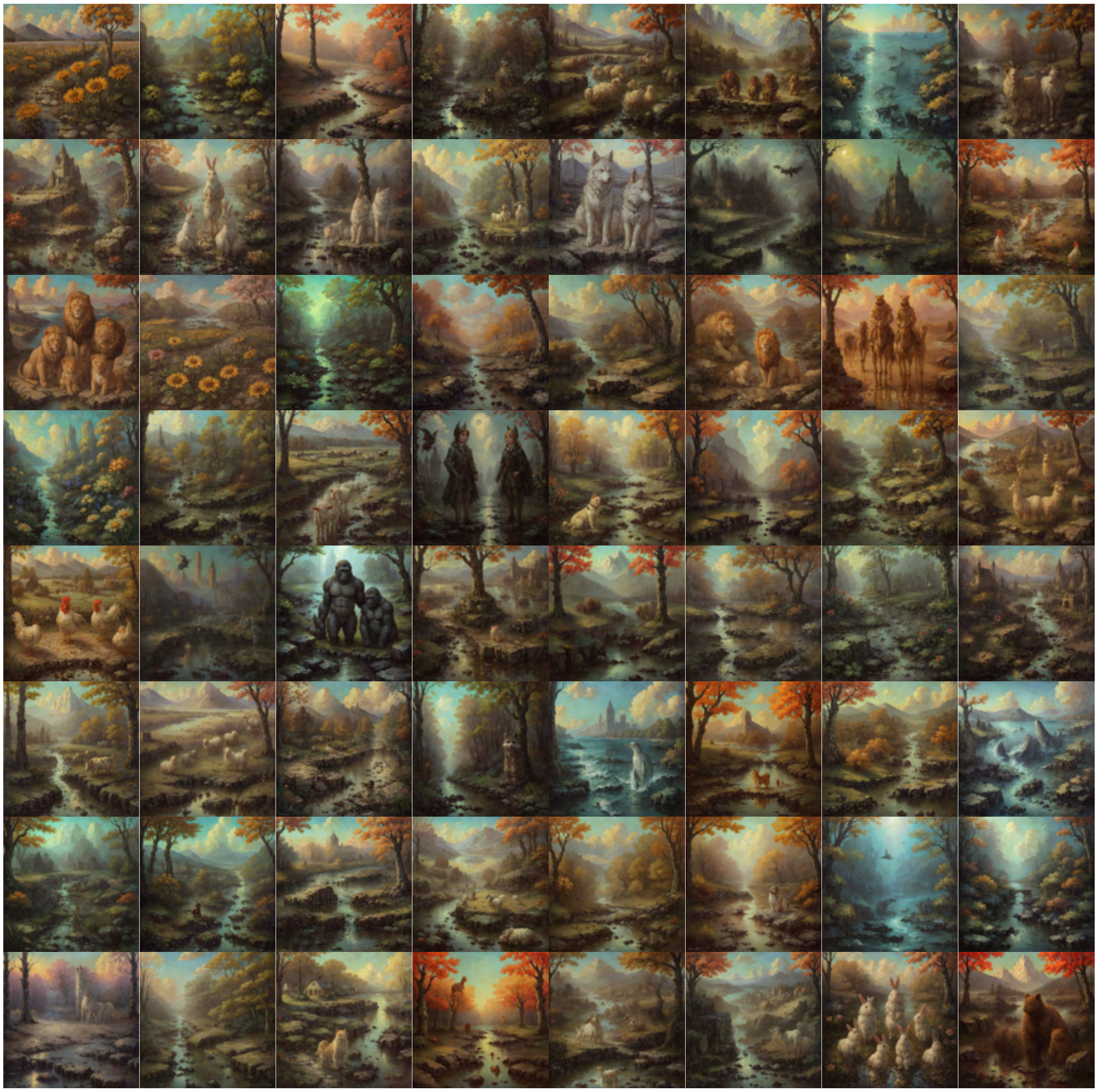


Figure 4: Generated images using PPO and REBEL during an intermediate checkpoint. We note that at the same number of epochs, REBEL observes a higher reward under the reward model. This can further be seen by the more diverse background of images generated from REBEL with less training time.

H.2.2 Example Generations of REBEL



I Ablation Analysis

η	Winrate (↑)	RM Score (↑)	KL($\pi\|\|\pi_{ref}$) (↓)
0.3	55.5%	1.37	10.4
0.7	59.9%	1.60	14.2
1.0	70.3%	2.44	29.2
2.0	62.5%	1.76	16.9

Table 3: REBEL ablation of the key hyperparameter η . The best-performing η for each metric is highlighted in bold.

Just like DPO, tuning REBEL is much more straightforward than PPO since the only hyperparameter REBEL introduced is η. We investigate how sensitive REBEL is to learning rate η in the loss. The results of ablation is shown in Table 3 with the same setting detailed in Appendix G.2 except for η. REBEL achieves the best performance when $\eta = 1$, while increasing or decreasing η leads to decreased performance. Our result here indicates that η is an important hyperparameter that requires tuning for achieving a good performance.

J Trade-off between Reward Model Score and KL-divergence

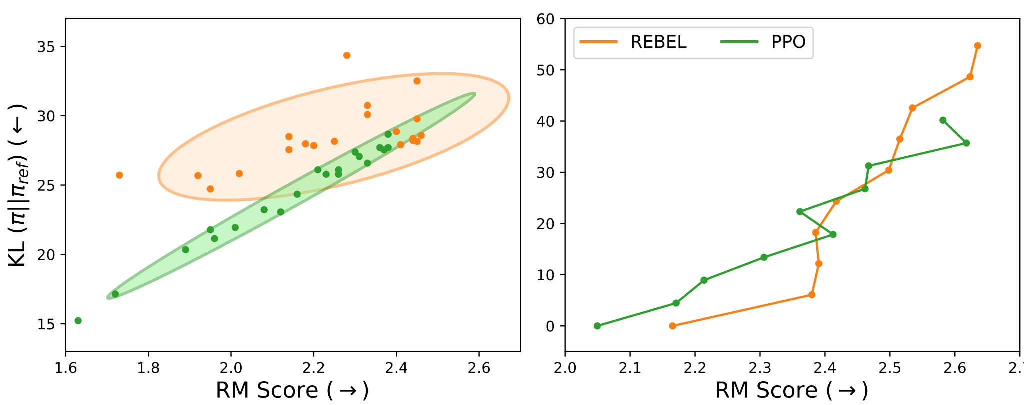


Figure 5: Plot of Reward vs KL-Divergence for 2.8B REBEL and PPO. We evaluate the models across the entire test set every 100 steps for 2,000 steps. Left: each point represents the average reward score and KL-divergence for a specific time step; the eclipse represents the confidence interval with 2 standard deviations. Right: we divide the KL distribution at the 2,000-step into 10 bins with equal size and average the corresponding RM scores in each bin.

The trade-off between the reward model score and KL-divergence is shown in Figure 5. We evaluate the 2.8B REBEL and PPO every 400 gradient updates during training for 8,000 updates. The sample complexity of each update is held constant across both algorithms for fair comparison. For the left plot, each point represents the average divergence and score over the entire test set, and the eclipse represents the confidence interval with 2 standard deviations. As observed previously, PPO exhibits lower divergence, whereas REBEL shows higher divergence but is capable of achieving larger RM scores. Notably, towards the end of the training (going to the right part of the plot), REBEL and PPO have similar KL and RM scores. For the right plot in Figure 5, we analyze a single checkpoint for each algorithm at the end of training. For each algorithm, we group every generation from the test set by its KL distribution into 10 equally sized bins and calculate the average of the corresponding RM score for each bin. We can see that REBEL achieves higher RM scores for generations with small divergence while requiring larger divergence for generations with the highest scores.

K Regression Loss During Training

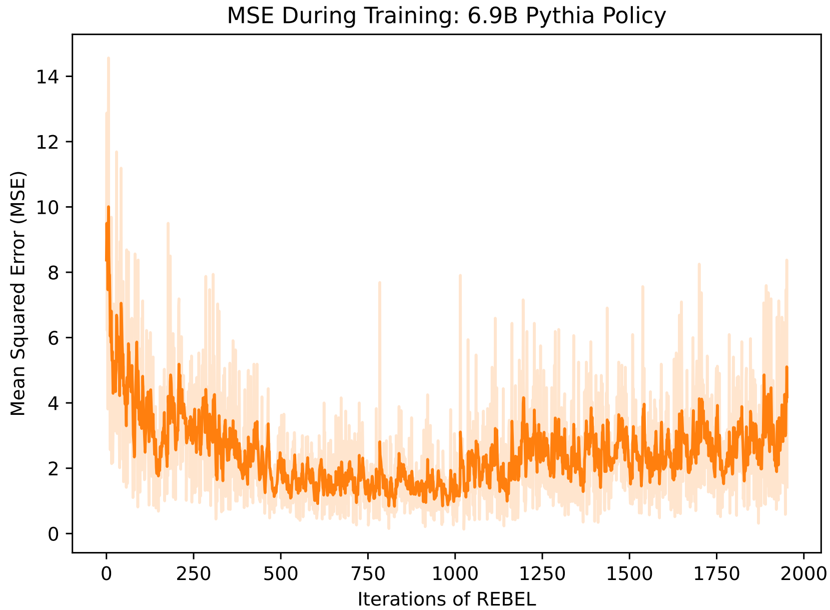


Figure 6: `REBEL`'s reward difference prediction error throughout training of our 6.9B parameter policy on the *TL;DR* task. The reward used for this task is unbounded with the range of values of the human labels in the validation set being $[-6.81, 7.31]$. We plot both the smoothed values with a moving average and the loss vales at each iteration.

Figure 6 shows the observed loss of Eq. 1 that we observed when finetuning the 6.9B Pythia model on *TL;DR*. We see that `REBEL` minimizes the loss throughout training maintaining a relatively low mean squared error given that our observed rewards were mostly between $[-10, 10]$. Note that our learned reward model, however, is unbounded.